

Name: _____

blank; insert 2 lines here

Directions: **Work only on this sheet** (on both sides, if needed); do not turn in any supplementary sheets of paper. There is actually plenty of room for your answers, as long as you organize yourself BEFORE starting writing.

1. () Here you will write an R S3 class "**stack**", do serve as a stack data structure. (Review: Items are *pushed* onto the stack, meaning here that they are added to the right end of a vector, and they are popped, meaning deleted from the right end.)

Here is a sample session:

```
> source("stack.R") # read in code
> ls()
[1] "pop" "push" "stack"
> stack("mystack") # make an object
> ls() # did we make the object?
[1] "mystack" "pop" "push" "stack"
> push(3,"mystack") # push 3
> mystack$data
[1] 3
> push(8,"mystack")
> mystack$data # check vector is now 3,8
[1] 3 8
> pop("mystack") # should return the 8
[1] 8
> mystack$data # check vector is now 3
[1] 3
```

The R functions **get()** and **assign()** do what their names imply, e.g.

```
z <- get("x",envir = .GlobalEnv)
```

copies the global variable **x** to **z**, and

```
assign("u",12,envir = .GlobalEnv)
```

is the equivalent of

```
u <<- 12
```

Fill in the blanks:

```
stack <- function(objname) {
  tmp <- # blank
  tmp$data <- vector()
  tmp$objname <- # blank
  # blank; insert 2 lines here
}
```

```
push <- function(val,objname) {
  tmp <- get(objname, envir=.GlobalEnv)
  # blank; insert 2 lines here
}
```

```
pop <- function(objname) {
  tmp <- get(objname, envir=.GlobalEnv)
  lng <- length(tmp$data)
  val <- # blank
```

Solutions:

```
# the stack class will (in this implementation) have objects only at the
# global level; an object has 2 member variables, the vector
# representing the stack, and the name of the object; values are pushed
# onto/popped off of the right end of the data vector
```

```
# constructor for "stack" class; objname is a string, namely a variable
# to be created at the top level, i.e. global
```

```
stack <- function(objname) {
  tmp <- list()
  tmp$data <- vector()
  tmp$objname <- objname
  class(tmp) <- "stack"
  assign(objname, tmp, envir = .GlobalEnv)
}
```

```
# pushes val onto the stack named objname
push <- function(val, objname) {
  tmp <- get(objname, envir = .GlobalEnv)
  tmp$data <- c(tmp$data, val)
  assign(objname, tmp, envir = .GlobalEnv)
}
```

```
# pops the stack named objname, returns the popped value
pop <- function(objname) {
  tmp <- get(objname, envir = .GlobalEnv)
  lng <- length(tmp$data)
  val <- tmp$data[lng]
  tmp$data <- tmp$data[-lng]
  assign(objname, tmp, envir = .GlobalEnv)
  return(val)
}
```